(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

USING MULTIPLE SEARCHERS
TO
LOCATE A RANDOMLY MOVING TARGET

by

Almir Garnier Santos

September, 1993

Thesis Advisor:	Robert F. Dell
Co-Advisor:	James N. Eagle

Approved for public release; distribution is unlimited.

Unclassified
Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified		1b Restrictive Markings	
2a Security Classification Authority		3 Distribution/Availability of Report Approved for public release; distribution is unlimited.	
2b Declassification/Downgrading Schedule			
4 Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol *(if applicable)* *OR	7a Name of Monitoring Organization Naval Postgraduate School	
6c Address *(city, state, and ZIP code)* Monterey CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey CA 93943-5000	
8a Name of Funding/Sponsoring Organization	6b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number	
Address *(city, state, and ZIP code)*		10 Source of Funding Numbers	

Program Element No	Project No	Task No	Work Unit Accession No

11 Title *(include security classification)* USING MULTIPLE SEARCHERS TO LOCATE RANDOMLY MOVING TARGETS

12 Personal Author(s) Almir Garnier Santos

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	23 Sep 93	56

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Search, Detection, Multiple Searchers, Constrained Path Planning, Team Effort, Moving Horizon, Genetic Algorithm, Branch and Bound,Search for moving objects

19 Abstract *(continue on reverse if necessary and identify by block number)*

The need to search effectively for objects presents itself in many civilian and military applications. This thesis develops and tests six heuristics and an optimal branch and bound procedure to solve the heretofore uninvestigated problem of searching for a Markovian moving target using multiple searchers. For more than one searcher, the time needed to guarantee an optimal solution for the problems considered is prohibitive. The heuristics represent a wide variety of approaches and consist of two based on the expected number of detections, two genetic algorithm implementations, one based on solving partial problems optimally, and local search. A heuristic based on the expected number of detections obtains solutions within two percent of the best known solution for each one, two, and three searcher test problem considered. For one and two searcher problems, the same heuristic's solution time is less than that of other heuristics considered. A Genetic Algorithm implementation performs acceptably for one and two searcher problems and highlights its ability, effectively solving three searcher problems in as little as 20% of other heuristic run-times.

20 Distribution/Availability of Abstract xx unclassified/unlimited ___ same as report ___ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Robert F. Dell	22b Telephone *(include Area Code)* (408) 656-2853	22c Office Symbol OR/De

DD FORM 1473,84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

Using Multiple Searchers
To
Locate A Randomly Moving Target

by

Almir Garnier Santos
Lieutenant Commander, Brazilian Navy
B.S., Brazilian Naval Academy, 1981

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September, 1993

Author: Almir Garnier Santos

Approved by: Robert F. Dell, Thesis Advisor

James N. Eagle, Co-Advisor

Alan R. Washburn, Second Reader

Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

The need to search effectively for objects presents itself in many civilian and military applications. This thesis develops and tests six heuristics and an optimal branch and bound procedure to solve the heretofore uninvestigated problem of searching for a Markovian moving target using multiple searchers. For more than one searcher, the time needed to guarantee an optimal solution for the problems considered is prohibitive. The heuristics represent a wide variety of approaches and consist of two based on the expected number of detections, two genetic algorithm implementations, one based on solving partial problems optimally, and local search. A heuristic based on the expected number of detections obtains solutions within two percent of the best known solution for each one, two, and three searcher test problem considered. For one and two searcher problems, the same heuristic's solution time is less than that of other heuristics considered. A Genetic Algorithm implementation performs acceptably for one and two searcher problems and highlights its ability, effectively solving three searcher problems in as little as 20% of other heuristic run-times.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To God above all and everyone.

To Sueli, my wife, for believing and continually convincing me that I was doing a superb job during this long, arduous but edifying journey.

To Almir Junior, for being the most understanding and grown up seven years old kid I have ever met.

To my relatives and friends for the spiritual support and comfort provided when nothing else seemed to work.

To my advisors, for the ever supporting and straight guidance during crucial design decisions of the project.

To Dr. Robert F. Dell for being more than a thesis advisor; for being a friend. Without him there would be not much to be proud of at the end.

To LCDR Gustavo H. Martins for the professional and friendly way of getting me acquainted with his previous work.

To the Brazilian Navy authorities for trusting this sailor in a so technically demanding deployment.

To all the people that I bothered in hallways talking about this "incredibly challenging problem" that I fell in love with, and who pretended to be interested.

EXECUTIVE SUMMARY

A. BACKGROUND

The need to search effectively for objects presents itself in many civilian and military applications. Civilian applications include the search for lost hikers, endangered animal species, and shoals of fish. Military applications are vast and were a driving force in the early days of Operations Research during World War II.

Effective models for search problems must cope with the behavior of a moving target, which is unknown in general, as well as searcher limitations, such as speed, endurance, detection range and precise navigation. This thesis develops and tests six heuristics (approximate methods) and a branch and bound procedure (guarantees an optimal solution) to solve the heretofore uninvestigated problem of searching for a randomly moving target using multiple searchers.

If the way a given target is expected to move and detection probabilities associated with friendly platforms when searching for such a target are obtainable, the algorithms developed in this thesis are useful. These algorithms recommend paths that, if followed by the friendly units, result in the maximum probability of detecting the target.

B. APPROACH

The problem of finding the best paths to follow for the maximum probability of detection is a very hard problem. This is because the number of alternatives grows quickly with problem parameters, such as the amount of time available to search and number of friendly searching units.

This thesis develops good approximate algorithms that avoid prohibitive run-times. This capability allows the algorithms to support real-time field requirements.

C. ACCOMPLISHMENTS

The various approximate algorithms developed in this thesis were tested against twenty seven problem instances.

One of the heuristics based on maximizing the expected number of detections obtains solutions within two percent of the best known solution for each one, two, and three searcher test problem ccnsidered. For one and two searcher problems, the same heuristic's solution time is less than that of other heuristics considered.

A Genetic Algorithm Heuristic performs acceptably for one and two searcher problems and highlights its ability, effectively solving three searcher problems in as little as 20% of other heuristic run-times.

I. INTRODUCTION

A. PROBLEM DEFINITION

The need to search effectively for objects presents itself in many civilian and military applications. Civilian applications include the search for lost hikers, endangered animal species, and shoals of fish. Military applications are vast and were a driving force behind the work of the Operations Evaluation Group in the early days of Operations Research during World War II [Ref. 1: preface].

Effective models for search problems must cope with the behavior of a moving target, which is unknown in general, as well as searcher limitations, such as speed, endurance, detection range and precise navigation. Aiming for generality, in a controlled way, many authors consider a randomly moving target where the dependency between "legs" of the target motion is Markovian. This thesis develops and tests six heuristics and an optimal branch and bound procedure to solve the heretofore uninvestigated problem of searching for Markovian moving targets using multiple searchers.

B. PROBLEM FORMULATION

This thesis extends the single searcher model proposed by Eagle and Yee [Ref. 2] to multiple searchers. Both single and multiple searcher models use discrete time with a single

target's motion modeled as a Discrete Time Markov Chain. The target is constrained to a single cell within a grid each time period, and has its movement alternatives, between time steps, restricted to adjacent cells. The initial probability distribution for the target and the target's Markovian transition matrix are assumed known.

The initial positions for the searcher(s) has to be specified. The searcher(s) has the same type of movement restriction as the target and a limited time to search. The search path's effectiveness is the cumulative probability of detection along the searchers' path(s) and detection occurs with a specified probability when the searcher and target occupy the same cell. Each time period, the probability distribution for the target throughout the area is Bayesian updated for non-detection.

An appropriate formulation for the multiple searcher problem, an extension of the single searcher problem of Eagle and Yee [Ref. 2], follows the introduction of appropriate notation.

1. Path Constrained Formulation (PDF)

Indices

i, i', k = cell,

j = searcher,

t = time step ($t = 1,2,\ldots,T$),

ω = path (where $\omega(t)$ is the cell occupied at time t).

Data

α_{ij} = detection rate in cell i, for searcher j. The probability of detection in a given cell is $1 - \exp(-\alpha_{ij})$,

Ω = set of all feasible target paths,

C_i = set of cells adjacent to cell i ,

p_ω = probability of target following path ω,

s_j = starting cell for searcher j at time zero.

Variables

$X_{i,\omega(t),j,t}$ = *One, if searcher j moves f:om cell i,* at time t-1, to cell $\omega(t)$ at time t and zero otherwise.

Formulation

Objective Function

$$Max\ (1 - \sum_{\omega\in\Omega} p_\omega \exp(-\sum_t \sum_j \alpha_{\omega(t)j} \sum_{[i|\omega(t)\in c_i]} X_{i,\omega(t),j,t}))$$

Subject to:

$$\sum_{i' \in C_{s_j}} X_{s_j, i', j, 1} = 1 \qquad , \ \forall j \qquad (1)$$

$$\sum_{i} \sum_{i'} X_{i, i', j, t} \leq 1 \qquad , \ \forall j \ , \ \forall t \qquad (2)$$

$$\sum_{[i \mid i' \in C_i]} X_{i, i', j, t-1} - \sum_{k \in C_{i'}} X_{i', k, j, t} = 0 \ , \ \forall i', \forall \ t>1, \forall j \ (3)$$

The formulation maximizes the probability of detection within the set of feasible paths Ω, subject to the constraints that:

1) Each searcher's initial search effort (t=1) must be in a cell adjacent to the starting position;

2) Each searcher can move at most once between time periods. Since the maximum objective function value is sought, the exclusion of this constraint could result in multiple paths for each searcher;

3) All search effort has to be done within the set of adjacent cells, at any time step, for any given searcher.

Changing PDF's objective function to maximize the expected number of detections (see Eagle and Washburn [Ref. 3]) for searchers who search "blind" until time T, provides an upper bound on the solution to PDF as shown by Martins [Ref. 4]. This also simplifies the PDF formulation since explicit enumeration of all possible paths is not needed. The

"principle of optimality" therefore holds and the problem can be solved as a shortest path problem. The simplified formulation (EDF) follows the introduction of appropriate notation.

2. Expected Number of Detections Formulation (EDF)

<u>Indices</u>

i, i' = cell,

j = searcher,

t = time step ($t = 1,2,\ldots,T$).

<u>Data</u>

Pd_{ij} = detection probability for searcher j in cell i $(1 - \exp(-\alpha_{ij}))$,

C_i = set of cells adjacent to cell i ,

PT_{it} = probability of target being in cell i at time t with no update for unsuccessful search $(P\{\omega(t) = i\})$,

s_j = starting cell of searcher j;

<u>Variables</u>

X_{ijt} = One, if cell i is visited by searcher j at time t and zero otherwise.

Formulation

Objective Function

$$Max \sum_{i} \sum_{j} \sum_{t} X_{ijt} Pd_{ij} PT_{it}$$

Subject to:

$$\sum_{i' \in C_{s_j}} X_{i'j1} = 1 \quad , \forall j \qquad (4)$$

$$\sum_{i} X_{ijt} \leq 1 \quad , \forall j \quad , \forall t \qquad (5)$$

$$X_{ijt-1} \leq \sum_{i' \in C_i} X_{i'jt} \, , \quad \forall i \quad , \forall j \quad , \forall \; t > 1 \qquad (6)$$

This formulation maximizes the expected number of detections along the path, subject to the constraints that:

4) Each searcher's initial search effort (t=1) must be in a cell adjacent to the starting position;

5) Only one cell can be assigned to each searcher during each time step;

6) Each searcher can only move to an adjacent cell.

C. PROBLEM DIFFICULTY

Trummel and Weisinger in their 1986 "The Complexity of the Optimal Searcher Path Problem."[Ref. 5] show that the path constrained search problem for a stationary target is NP-complete. An example highlights the problem's complexity. A single searcher using five time steps to search a nine cell problem has approximately 1,024 feasible paths to choose from. The same problem with 10 time steps has about 1,048,576 feasible paths. This problem with three searchers has about 1.15×10^{18} feasible paths. The path constrained search problem with multiple searchers is at least as hard, and by being so, the main thrust of this thesis is the development, implementation, testing and evaluation of relatively fast and robust heuristics. These heuristics should be well suited for practical applications like tactical decision aids.

D. THESIS OUTLINE

This thesis develops, analyzes, and tests six heuristics for the multiple searcher path problem: two are extensions of the heuristics proposed by Martins [Ref. 4], a genetic algorithm, a hybrid genetic algorithm that incorporates other heuristics, a heuristic based on solving partial problems optimally, and a local search method. This thesis also develops an optimal branch and bound procedure extended from Martins [Ref. 4].

The specific organization of this work is as follows. Chapter II presents a literature survey on related problems. Chapter III details each heuristic. Chapter IV provides detailed computational comparisons between the heuristics applied to a set of test problems using one, two, and three searchers. Finally, Chapter V presents conclusions.

II. PROBLEM BACKGROUND

The operations research literature contains numerous books and published articles on stationary target problems. The consensus of the research community is that the framework for these problems was laid down by the United States Navy Antisubmarine Warfare Research Group in 1942 in response to the Atlantic German submarine threat [Ref. 1]. Subsequent work by many researchers took the stationary target problems into a mature state where solutions are available for the most common problems and improvements are hard to find [Ref. 6].

The case of a lone searcher looking for a single moving target has also been widely studied and can be divided into two major classes: Two-sided search and One-sided search.

Two-sided search problems are concerned with the possibility that the target is aware that a search effort is being carried out against him and attempts to avoid detection or capture. Game Theory is the natural tool here (see Thomas and Washburn [Ref. 7], and Eagle and Washburn [Ref. 3]). One-Sided search problems assume either the target is not aware of the search or the target needs to accomplish its own task and it is not willing to evade the searcher. Through this reasoning the idea of a Bayesian probability distribution and update of the target's position is straightforward. The One-sided search problems are usually further divided as Optimal

Density or Optimal Path Problems. Both groups in more recent work have dealt with the target motion being modeled as a Discrete Time Markov Chain and the "continuous search" in each time step being modeled by an exponential law of detection.

Optimal density problems tend to be easier problems than optimal path problems since integrality or adjacent movement constraints can be dropped. These problems are well suited to situations when the searcher and target speeds differ by more than an order of magnitude. Brown [Ref. 8] made important progress in optimal density problems by developing an algorithm that solves the moving target problem as a sequence of stationary target problems. Washburn [Ref. 9] gave a counterpart algorithm for the discrete search effort case as did Stone et. al [Ref. 10].

Optimal path problems with the characteristics described above are tackled by Stewart [Ref. 11,12] using an optimal branch and bound procedure. Eagle's branch and bound approach [Ref. 2] was first to obtain bounds by using the Frank-Wolfe algorithm to solve a problem where integrality restrictions are relaxed. Martin's branch and bound algorithm [Ref. 4] uses the maximum expected number of detections to provide bounds.

Another interesting model for Optimal Path Problems is the continuous time and space case where the constraints on the searcher's motion are given by a set of differential equations

that the searcher's path has to obey. Ohsumi [Ref. 13] is a good example of such a model.

According to Weisinger et al [Ref. 14] in their survey, 125 references are available for one-sided search problems and 61 to search games but none are listed for the multiple search problem or team effort under the same modeling assumptions (the subject of this thesis).

III. ALGORITHMS

Seven algorithms (six heuristics and one exact procedure) are developed to determine the path that maximizes the probability of detecting a randomly moving target using multiple searchers. This chapter introduces the network structure common to all algorithms and then describes each algorithm using pseudocode.

Eagle and Yee [Ref. 2] use a network structure with nodes or cells as locations where the searcher can allocate his effort during one time step. If the searcher is in cell i at time t, at time t+1 he can only search cells adjacent to cell i (denoted as C_i). The state of this system can be represented by a sequence of cells/nodes from time one (t=1), until the last time step available to the searcher (t=T). The searcher's effort results in a feasible path, ω. The objective of the problem is to find a path that, if followed, maximizes the probability of detecting the target.

The multiple searcher version, developed here, employs the same network structure with an expanded state space to account for extra searchers. The difference is explained using an example. Suppose two searchers are initially stationed at cell 1 and $C_1 = \{1,2,4\}$. At the next time step (t=1), each searcher has a separate choice for the next move resulting in the possible combined states of {(1,1), (1,2), (1,4), (2,1),

(2,2), (2,4), (4,1), (4,2), or (4,4)}; where the first (second) entry is the location of searcher 1 (searcher 2). Table I shows the effect of increasing the number of searchers.

Table I State Space Examples

Number of Searchers	**Starting Position**	**Possible Positions (States) at Time One**
1	(1)	(1), (2), or (4)
2	(1,1)	(1,1),(1,2),(1,4),(2,1),(2,2),(2,4),(4,1),(4,2)or(4,4)
3	(1,1,1)	(1,1,1),(1,1,2),(1,1,4),(1,2,1),(1,2,2),...,(4,4,4)

A. DESCRIPTION OF HEURISTICS

1. Local Search (LS)

Local search (see Papadimitriou and Steiglitz [Ref. 15]) is a basic approach used to solve combinatorial optimization problems. This thesis includes it as a benchmark of how well a simple heuristic performs on our test problems. An implementation of local search with random restarts applied to the multiple searcher problem is easily explained using the pseudocode below. The pseudocode employs the notation PD for probability of detection.

1 Repeat

2 Create a feasible searcher path, ω_{old}

3 Compute PD_{old} of ω_{old}

4 PD_{best} <- PD_{old} , ω_{best} <- ω_{old}

5 For t = 1 to (T-2) Do

6 ω_{new} <- ω_{old}

7 For each cell $i \in C_j$ where $j=\omega_{new}(t)$

8 $\omega_{new}(t+1)$ <- i

9 For each cell $i' \in C_i$

10 $\omega_{new}(t+2)$ <- i'

11 If ω_{new} feasible compute PD_{new}

12 If PD_{new} >= PD_{old}

13 PD_{old} <- PD_{new} , ω_{old} <- ω_{new}

14 next i'

15 next i

16 If PD_{old} >= PD_{best} go to step 4

17 Until exceed number of restarts or time limit

18 Return path that yielded the highest PD

2. Heuristic_1 (H1)

Martins [Ref. 4] develops a heuristic (Heuristic_1) based on the expected number of detections. The redefined network structure allows this heuristic to also be used for multiple searchers. The pseudocode below employs the shorthand Path(t) to store the cells occupied by the searchers

at time t on the path chosen by H1 and ED for the expected number of detections.

```
Path(0) <- Initial cell of searchers
For t = 1 to total time steps available (T) Do
    Let Path(t-1) be the searchers' cell
    Find path ω maximizing ED for t,...,T
    Path(t) = ω(t)
    Update the probability mass for the target
Compute PD when searchers follow Path
Return Path and PD.
```

The complexity of this algorithm is O((Number of Cells)$(C_i)^N(T)^2$), where N is the number of searchers.

3. Heuristic_2 (H2)

Another heuristic (Heuristic_2) developed by Martins is easily extended to incorporate multiple searchers. This heuristic expands on H1 by basing the next node added to the searchers' path on more than the path with the maximum ED. Specifically, H2 generates a path for every possible single next move, extends the path to T using the maximum expected number of detection criterion, and picks the path with the largest probability of detection. The pseudocode below fully explains H2 using PD(ω_i) for the probability of detection associated with path ω_i.

Path(0) <- Initial cell of searchers

For t = 1 to Total time steps available (T) Do

Let Path(t-1) be the searchers' cell

For all cells i $\in C_{Path(t-1)}$ Do

    Find path ω_i maximizing ED for t+1, ..., T,

    where $\omega_i(t) = i$

    Compute PD(ω_i)

Path(t) = k such that PD(ω_k) = Maximum_i PD(ω_i)

Update probability mass of target given $\omega_k(t)$

Compute PD following Path

Return Path and PD.

This algorithm complexity is O((Number of Cells^2)$(C_i^2)^N(T)^2$).

4. Genetic Algorithm (GA)

Genetic Algorithms (see Goldberg [Ref. 16] and Holland [Ref. 17]) are self improving algorithms that work by means of natural selection, or survival of the fittest. A crude implementation of a genetic algorithm to the multiple searcher problem provides an introduction to basic operators and characteristics of such algorithms. Each step of this crude implementation is then expanded into the form used for the computational work reported in this thesis.

```
Randomly create a population of n feasible paths (Ω_old)
Ω_new <- φ
For generation=1 to maximum number of generations
    While insufficient number of paths in Ω_new
        Select two paths (ω_1,ω_2) from Ω_old
        Apply Cross-Over on ω_1 and ω_2 to form new path
        Apply Mutation operator on the new path
        Calculate PD for the new path and add to Ω_new
    Ω_old <- Ω_new, Ω_new <- φ
Return path that yielded the highest PD
```

One of the characteristics of Genetic Algorithms is the need to set run-time parameters, such as the population size, the probability of cross-over, the probability of mutation, and the number of generations. This painful process is automated in the Genetic Algorithm implementation of this thesis. Values described below are empirically chosen to be robust across a variety of problems which may limit the efficiency of the algorithm for particular cases.

The Genetic Algorithm literature refers to a population as an ordered collection of problem variables (in our case paths) and its associated fitness (PD). We allow the population size to vary between generations subject to a

maximum respectively of n = 200, 400, or 600 paths for the one, two, three searcher problems. (Again, all run-time parameters are empirically derived to provide good performance across a wide variety of problems.) The size of the initial population is calculated as:

$$C1 * \lfloor C2 * \ln(Number\ of\ Cells) * \sqrt{T} \rfloor; \qquad (7)$$

where C1 and C2 are constants that take the values one and seven for the one searcher problem, three and five for the two searcher problem and six and five for the three searcher problem.

Careful creation of the initial population ensures that every path in that population is distinct. As a general rule, the more diversity that exists within a population the greater ability of Genetic Algorithms to evolve in improving directions. To improve the best path in the initial population, the LS heuristic (steps 5 to 15 only) is used on the path.

The new population size adjusts based on the following statistic of the previous population:

$$\lfloor (\frac{(0.5 * (\sqrt{(\frac{PD_{best}}{2})} - \sqrt{(\frac{PD_{worst}}{2})})}{PD_{average} + \epsilon} - 1) * 10 \rfloor; \qquad (8)$$

where ϵ is a very small number. The value of the statistic is added to the initial population size. This simply computed statistic provides an indication of how skewed the population

is towards the more likely poor paths. (Of course, to calculate the actual skewness coefficient a time consuming step of finding the median would have to be conducted.) Adding its value to the initial population size increases (decreases) the number of paths when the population tends toward poor (good) paths.

Step 5 of the GA pseudocode selects paths from the old generation to create the new generation. A simple and widespread way of picking individuals is by "roulette wheel selection" which advocates randomness but controlled so that only the fittest survive. This concept is applied by randomly picking individuals with probability equal to the $\sqrt{}$(PD of the path) divided by the sum of $\sqrt{}$(PD of all the paths in its population). The square root function allows enhanced discrimination between very good and very bad paths but reduces discrimination between paths with high PD. Within any new generation the three best paths from the previous generation are left unchanged guaranteeing the best path encountered so far survives. The Genetic Algorithm literature [Ref. 18] refers to this process as elitism.

The most important operation to be executed on the newly selected paths is the crossover (step 6) which probabilistically creates a random mix of the two parent paths (perhaps leaving a parent intact). This is done in an attempt to obtain good characteristics of both paths used to create the child. It is a random mix since the time where the

crossing takes place is uniform and randomly chosen between one and T. Specifically, given the crossing time t' and two parents ω_1 and ω_2, the child is ω where

$$\omega(t) = \omega_1(t) \text{ for } t \le t', \text{ and}$$

$$\omega(t) = \omega_2(t) \text{ for } t > t'.$$

For example, suppose in a single searcher problem, that two paths are selected (1->2->3 and 1->4->5) and that time two is randomly chosen. The resulting path, if feasible, is 1->2->5.

As in life, every new born child has the opportunity to evolve by acquiring characteristics that are not inherited from his or her parents. The mutation operator (step 7) serves this role by probabilistically changing path cells. However, guaranteeing feasible mutations is not easy for path constrained problems. Consider the resulting path from the example above, 1->2->5, and let $C_1 = \{4,2,1\}$, $C_2 = \{5,3,2,1\}$, $C_4 = \{1,5,7\}$ and $C_5 = \{8,6,5,4,2\}$. The mutated value at the second location on the path (currently 2) must be contained in both C_1 and C_5. Possible values are therefore only 4 and 2.

As generations progress, decreasing the probability of cross-over and increasing the probability of mutation, which is the probability that the cross-over (mutation) operator is used, empirically improves the Genetic Algorithm's performance. An initial high cross-over probability provides for a diverse population. Increasing the probability of mutation helps avoid the tendency of the best individual converging to a local optimal. The initial probability of

cross-over (mutation) is 0.5 (0.2). The probabilities are adjusted each generation in the following three ways which, in the order presented, provide a continuous change, a change based on convergence, and a change based on the potential of the previous population:

- The addition (subtraction) of 0.03 to the mutation (crossover) probability,
- The addition (subtraction) of NumberOfReps/20 to the mutation (crossover) probability where NumberOfReps equals the number of generations having the same best path,
- The addition (subtraction) of the result of equation (9) to the probability of crossover (mutation) where PD_{best}, PD_{worst}, $PD_{average}$ are taken from the previous population and C3 equals 10 (20):

$$\frac{\ln\left(\dfrac{\left(0.5 * \left(\sqrt{\left(\dfrac{PD_{best}}{2}\right)} - \sqrt{\left(\dfrac{PD_{worst}}{2}\right)} \right)\right)}{PD_{average} + \epsilon} \right)}{C3} \qquad (9)$$

After these three terms are algebraically added to the previous probability of cross-over (mutation), the final value, if outside the bounds for the operator, is then rounded to the appropriate interval limit. The upper and lower limit of cross-over (mutation) are 0.8 (0.8) and 0.4 (0.1), respectively.

Every three generations, a further attempt to diversify and improve the population of paths is made. It is explained in the pseudocode below.

```
1  If NumberOfRep < seven then
2      If generation is even
3          Apply LS on best path
4      else
5          Apply LS on second best path
6  else
7      If generation is odd
8          Apply LS on third best path
9  If NumberOfRep > ten
10     Apply LS on forth best path
11 If (top 10% of paths are not distinct)
12     Replace nondistinct paths
       probabilistically with a new random path.
```

Step 12 above replaces replicated paths according to a Diversity Parameter. The Diversity Parameter in this implementation has an initial probability value of 0.5 and upper (lower) bound of 0.7 (0.2). Its value is adjusted using equation (9) with C3 = 10.

Another step taken to diversify the population, aimed at the poorest group of paths, is to randomly generate a new path to replace any path that shows PD less than 20% of the best path. Equation (9) with C3 = 10 adjusts the value of 20% within 10% and 30% between generations.

When the NumberOfRep is greater than 11 (the best solution has not changed for 11 generations), the first one

third of the population is decimated and replaced by random paths. Decimation takes place only once in our implementation and the NumberOfRep is discounted four after its application.

A number of stopping conditions terminate the GA heuristic when exceeded: maximum number of generations (Number of Searchers * 100), maximum amount of run-time (Number of Searchers^2 * ln(T) * ln(Number of Cells) * 3 minutes), NumberOfRep greater than twenty.

5. Hybrid Genetic Algorithm (HGA)

The HGA algorithm is the GA which includes in the starting population the three heuristic solutions produced by H1, H2 and the path that provided the maximum expected number of detections.

6. Moving Horizon (MH)

The MH algorithm uses divide and conquer, one of the three basic solution paradigms. The MH heuristic is based on empirical and theoretical evidence which suggests it requires significantly more than twice the time to solve the same problem having ten time steps compared to five time steps. Our MH heuristic breaks the true problem into subproblems (problems consisting of less time steps) which are optimally solvable within a reasonable amount of computer time. Empirically, aiming to get very good solutions without expending an unacceptable amount of computer time, this

implementation computes the horizon length(H) by means of equation (10).

$$\lfloor \ln(\textit{Number of Cells}) + \sqrt{T} \rfloor; \tag{10}$$

The horizon length is limited to eleven for the single searcher and to six for the two-searcher problem.

The MH pseudocode uses $\text{Path}_{IS}(t)$ and $\text{Path}_{O}(t)$ for cells contained on the path of the initial solution and partial optimal solution respectively.

```
1  Compute horizon length (H)
2  Compute initial Solution (IS) using H1
3  For k = 1 to T-H
4      Solve the subproblem for t = k to k+H
       optimally
5      If PD = 0 then
6          Path(k) = Path_IS(k)
7      else
8          Path(k) = Path_O(k)
9      Update the Probability mass for the target;
10 Compute PD when searchers follow Path;
11 Return Path and PD.
```

B. OPTIMAL BRANCH AND BOUND ALGORITHM

Using the redefined network, the branch and bound algorithm of Martins [Ref. 4] solves multiple searcher scenarios optimally. The algorithm is $O((C_i^N)^T)$ thus being of

limited use for most practical applications. Nevertheless, it is a reference against which the precision of the heuristics can be measured for most single searcher and some multiple searcher problem instances presented in Chapter IV.

IV. IMPLEMENTATION

All algorithms are implemented in Pascal and run on a 486/33 personal computer. This hardware choice is likely to be in any probable user's inventory. Another issue favoring the choice of this machine is greater precision offered for ordinary data types using the Borland Pascal 7.0 compiler. The Pascal compiler currently available on the Naval Postgraduate School mainframe (the machine used by Martins) has five less digits of precision for type Real and three for the type Double. The precision affects random number streams used intensively by GA. With less precision, the streams of pseudo-random numbers become more correlated thus degrading the performance.

The test problems investigated in this thesis are the same 9, 25, and 49 cell problems presented in Martins [Ref. 4]. The initial position for target and searcher(s) and the searcher(s) probability of detection are also as in Martins [Ref. 4]. When multiple searchers are present, their initial position is the same. The target motion is the "wandering around type." This motion, implemented as a discrete time markov chain, mimics the motion of ballistic submarines on patrol or polar bears looking for food. Of the problems available in the literature, this target motion seems to provide the greatest algorithmic challenge. See Eagle and Yee

[Ref. 2] for a description of how the transition matrix is derived.

Four basic characteristics (number of searchers, number of cells, number of time steps, and each searcher's probability of detection) are varied to values shown in Table II. Possible permutations of the number of cells, time steps, and probability of detection produces 27 test problems (shown in Table III) for each algorithm to solve with each number of searchers.

Table II Test Problem Variables

Searchers	Cells	Time Steps	Pd
1	9	4	0.33212
2	25	12	0.63212
3	49	20	0.93212

Our implementation allows more than one searcher to search the same location at the same time. The random search law is used, which allows the detection rates of the N searchers to be added. For example, suppose searcher 1 and 2 are searching the same cell at the same time step and searcher 1 (2) has a 0.5 (0.8) probability of detecting the target given they are both in the same cell. Then the detection rate for searcher 1 (2) is $\alpha 1$ ($\alpha 2$) where $1-e^{-\alpha 1}=0.5$ ($1-e^{-\alpha 2}=0.8$). The total detection rate becomes $\alpha=\alpha 1+\alpha 2$ or $0.69+1.61=2.30$ and the overall probability of detection is $1-e^{-\alpha}=0.90$.

Table III Problem Numbers

Problem	Cells	Time Steps	Pd
1	9	4	0.33212
2	9	4	0.63212
3	9	4	0.93212
4	9	12	0.33212
5	9	12	0.63212
6	9	12	0.93212
7	9	20	0.33212
8	9	20	0.63212
9	9	20	0.93212
10	25	4	0.33212
11	25	4	0.63212
12	25	4	0.93212
13	25	12	0.33212
14	25	12	0.63212
15	25	12	0.93212
16	25	20	0.33212
17	25	20	0.63212
18	25	20	0.93212
19	49	4	0.33212
20	49	4	0.63212
21	49	4	0.93212
22	49	12	0.33212
23	49	12	0.63212
24	49	12	0.93212
25	49	20	0.33212
26	49	20	0.63212
27	49	20	0.93212

A fictitious scenario fitting the test problem descriptions follows. Suppose a diesel submarine is observed by an S-3 Viking during an Anti-Submarine Warfare (ASW) operation to screen the USS "Eisenhower" as she transits to South Africa. The submarine's course would take it into the territorial waters of Brazil. The time and coordinates of the contact are transmitted to concerned authorities in Brazil. The Brazilian Navy opts for dispatching a Search and Attack Unit (SAU) composed of two Frigates ("Independência" and "Uniao").

The area of interest is divided in 25 cells according to the sensors' performances, distances involved, and the transition matrix chosen represents a "wandering around" type of target motion. Figure 1 illustrates the scenario.

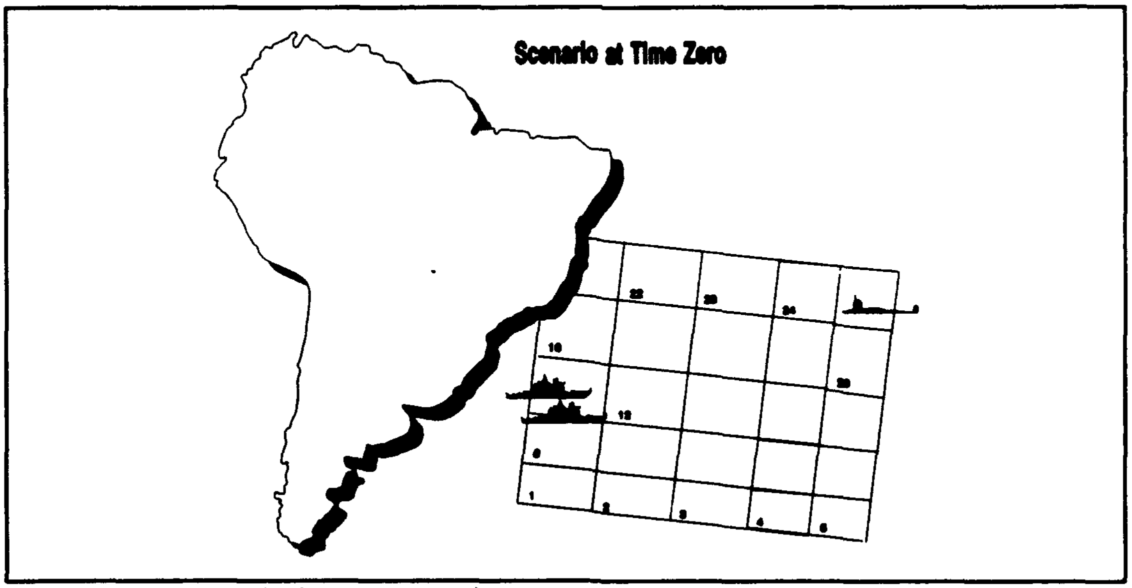

Figure 1 Possible Scenario

A. COMPUTATIONAL PERFORMANCE

1. One Searcher Results

Figure 2 shows the probability of detection achieved by the six heuristics and the optimal branch and bound procedure for each test problem using a single searcher. This

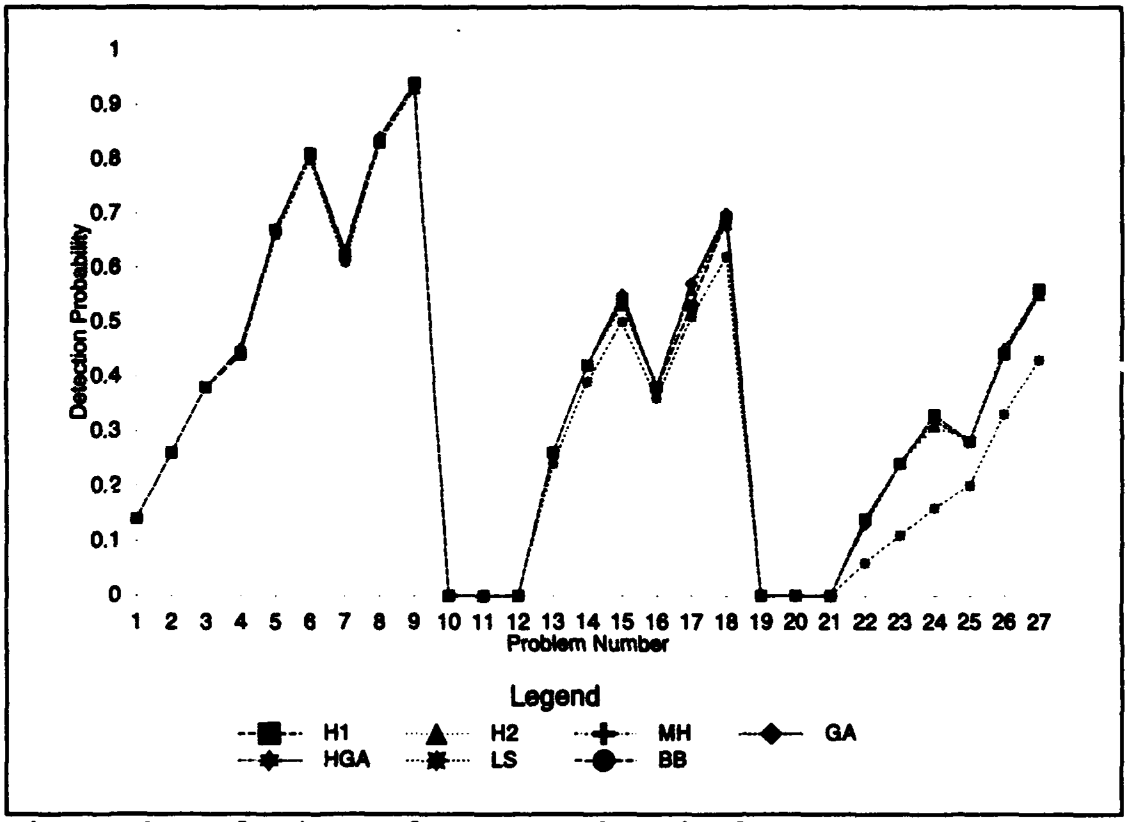

Figure 2 Solution Values For The Single Searcher Problem.

figure highlights the superior performance of all the heuristics when time is not considered. The only exception being LS which is included as an indication of how well a simple heuristic performs. Table IV shows the percentage away from the best known solution achieved by each heuristic.

Table IV Percentages Away From The Best Known Solution Obtained By Each Heuristic For The Single Searcher Problems. Starred problem numbers indicate optimal solutions.

Problem	H1	H2	MH	GA	HGA	LS
1*	0	0	0	0	0	0
2*	0	0	0	0	0	0
3*	0	0	0	0	0	0
4*	1	0	0	0	0	2
5*	1	1	0	1	0	2
6*	0	2	0	1	0	2
7	0	1	0	1	1	3
8	1	1	0	1	1	2
9	2	1	0	1	0	1
10*	0	0	0	0	0	0
11*	0	0	0	0	0	0
12*	0	0	0	0	0	0
13*	0	1	0	1	0	9
14*	0	3	0	0	0	9
15*	1	4	0	0	1	8
16	1	1	0	1	1	7
17	1	3	0	1	1	12
18	1	3	0	1	1	11
19*	0	0	0	0	0	0
20*	0	0	0	0	0	0
21*	0	0	0	0	0	0
22*	0	0	0	1	0	55
23*	0	2	0	2	0	53
24*	0	7	0	2	0	51
25	1	1	1	0	1	30
26	1	2	0	1	1	27
27	2	3	0	2	2	25

Run-times differ significantly as shown in Figure 3 where BB is limited to 60 minutes. LS run times are limited to 15 minutes or until a number of restarts exceeds the population size of GA.

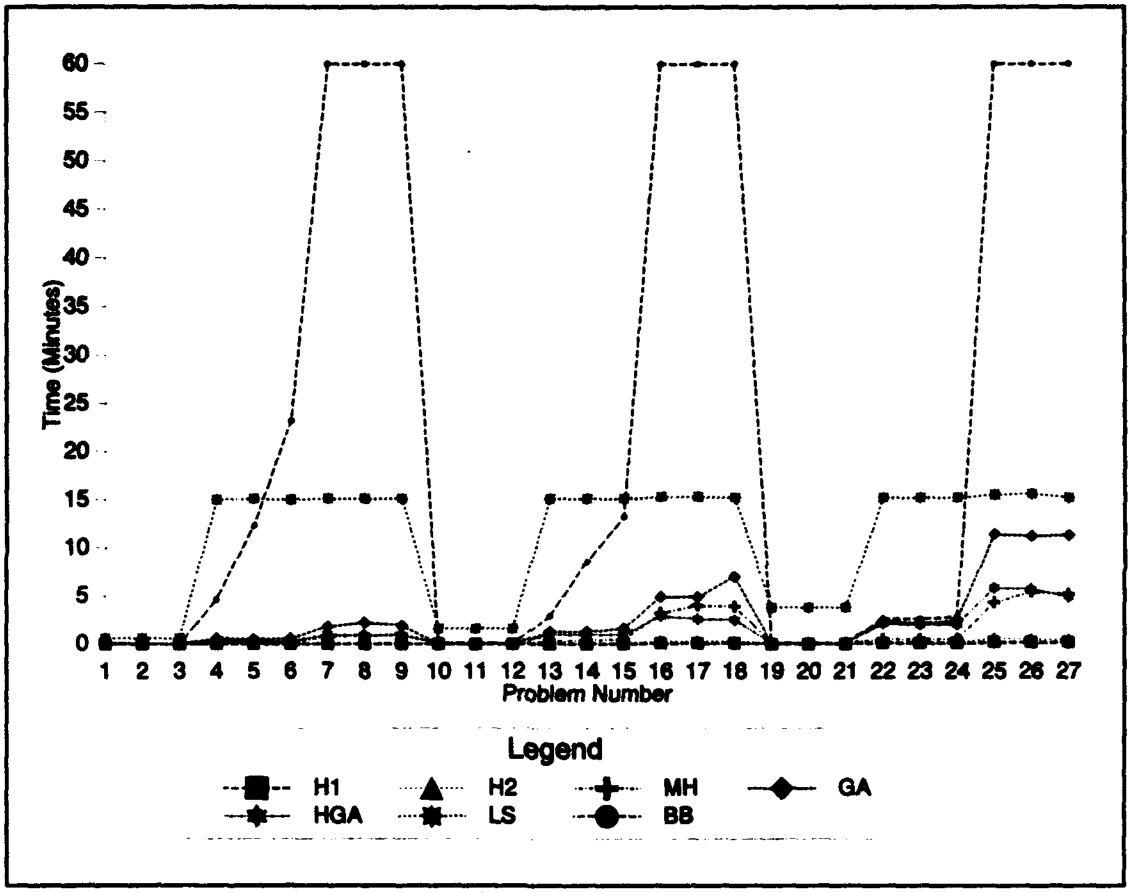

Figure 3 Run-Times For Single Searcher Problems. The BB is limited to 60 minutes. LS run-times are limited to 15 minutes.

Figure 4, presents run-times from zero to ten minutes to better illustrate the differences amongst the various heuristics.

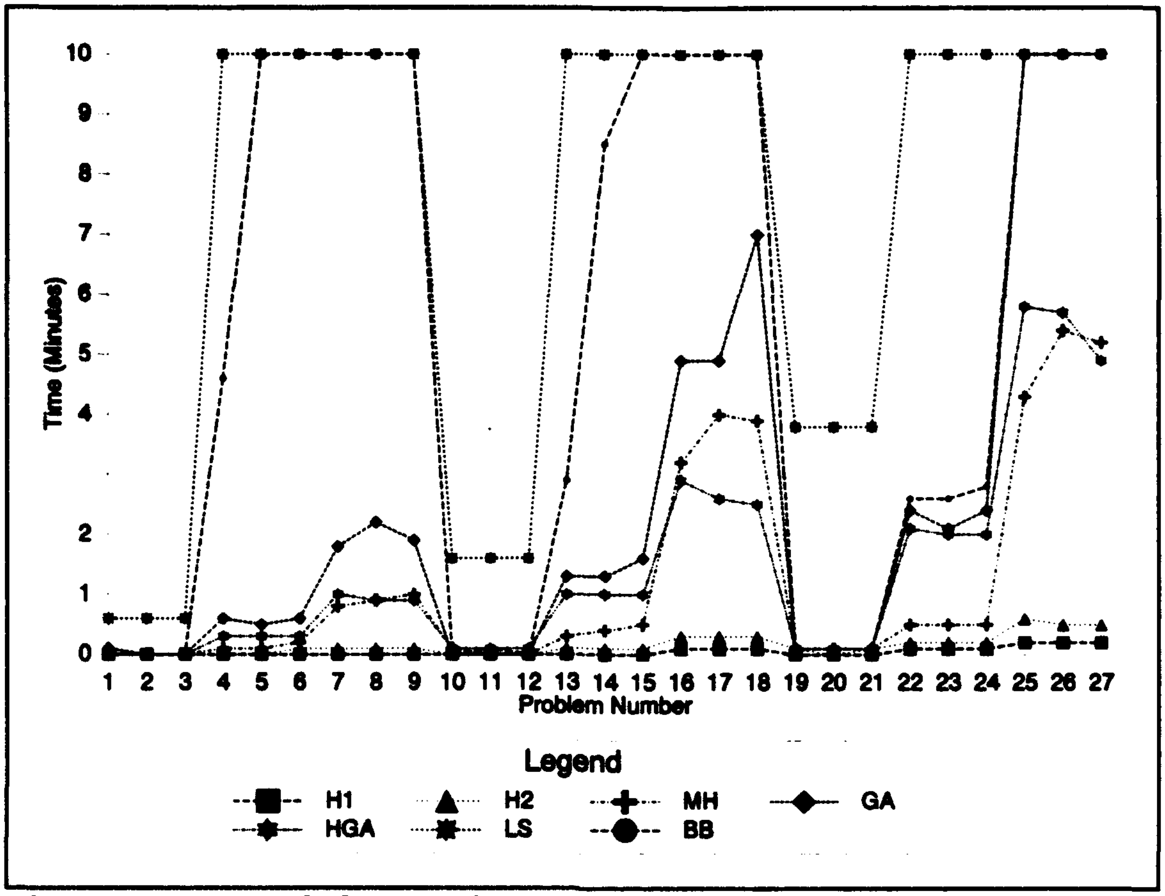

Figure 4 Expanded Run-Times For Single Searcher Problems

From Figure 4 and Table II it is clear that H1 outperforms the other heuristics in run-time and always obtain a solution within two percent of the best known. The MH heuristic uses a maximum of 11 time steps for the horizon which provides superior performance as indicated in Table II but with increased run-time as indicated in Figure 4.

2. Two Searcher Results

The same set of 27 problems is solved for the two searchers case. The complexity of this instance grows exponentially with the number of searchers. However, it gives

more insight on the capability of each of the algorithms proposed here to deal with real world problems. Figure 5 shows results achieved by each individual heuristic. Optimal solutions are not obtained due to excessive computational requirements. The nine cell problem with eight time step and two searchers had to run for almost five days before obtaining the optimal solution.

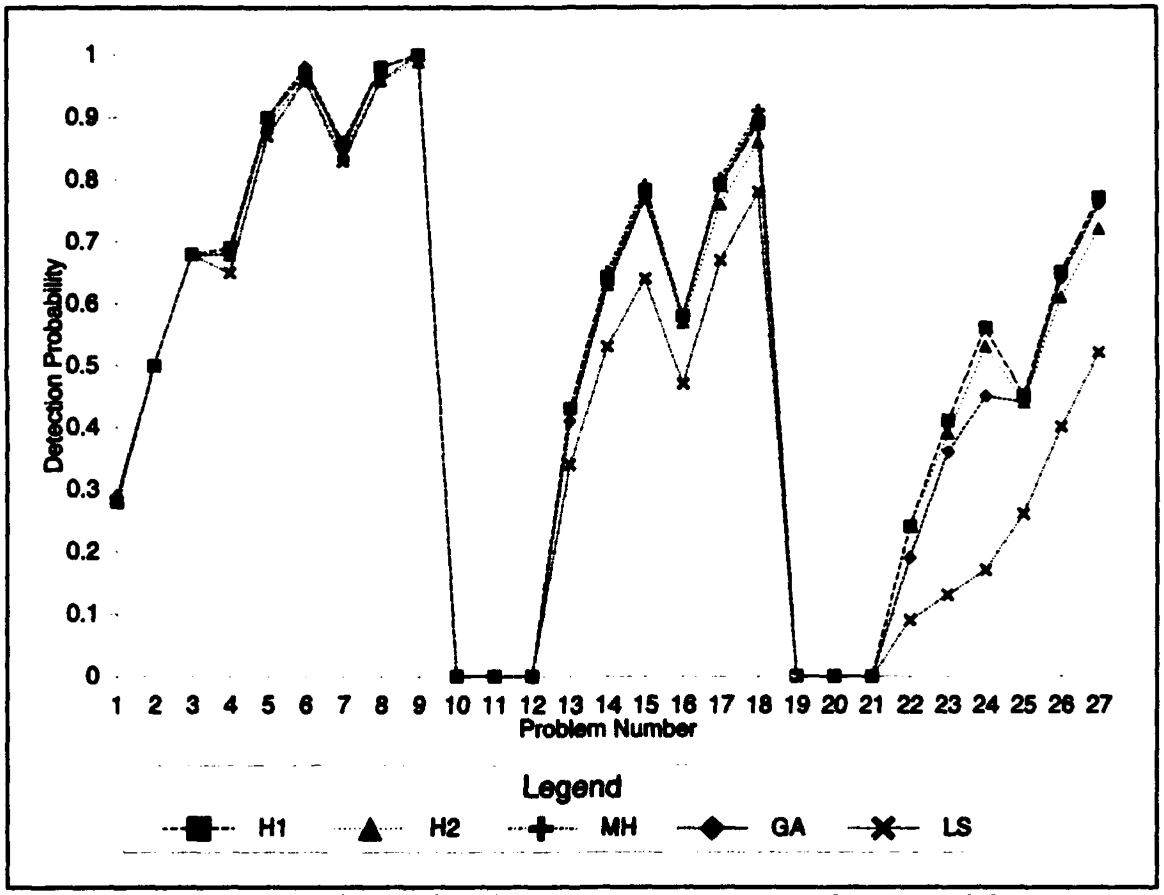

Figure 5 Heuristic Solutions For Two Searcher Problems.

Table V provides detailed information on heuristic performance and clearly shows MH provides superior solutions.

Table V Percentages Away From The Best Known Solutions Obtained By Each Heuristic For Two Searchers Problems.

Problem	H1	H2	MH	GA	LS
1	0	0	0	1	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	2	0	2	6
5	1	2	0	1	3
6	1	3	0	1	3
7	0	2	0	1	4
8	1	2	0	1	2
9	1	1	0	1	1
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	1	0	4	21
14	1	3	0	4	18
15	2	3	0	3	20
16	1	4	0	1	20
17	2	5	0	2	16
18	2	6	0	1	15
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	1	0	0	32	65
23	0	7	0	13	70
24	0	6	0	20	69
25	0	2	0	2	42
26	0	7	0	3	40
27	0	7	0	2	34

However, Figure 6 illustrates that the run-time necessary to obtain these results is significant.

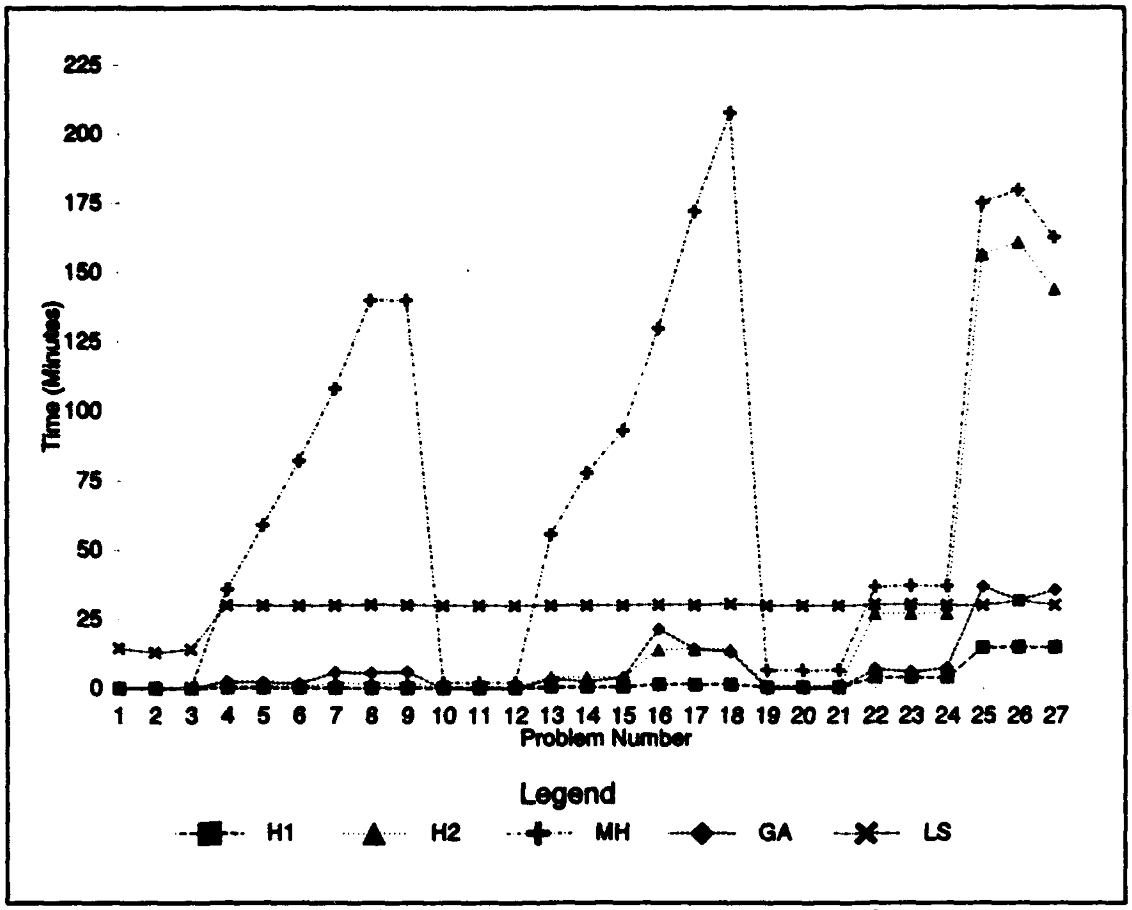

Figure 6 Running Times For Two Searcher Problems.

The solution times for problems 19 to 27 are distorted due to a limitation of the Borland Pascal 7.0 compiler which does not allow single data structures to exceed 64k. A number of programming changes are conducted to overcome this limitation which results in slower execution.

Once again the lower end of the time scale is expanded to better illustrate the running times of the fastest algorithms.

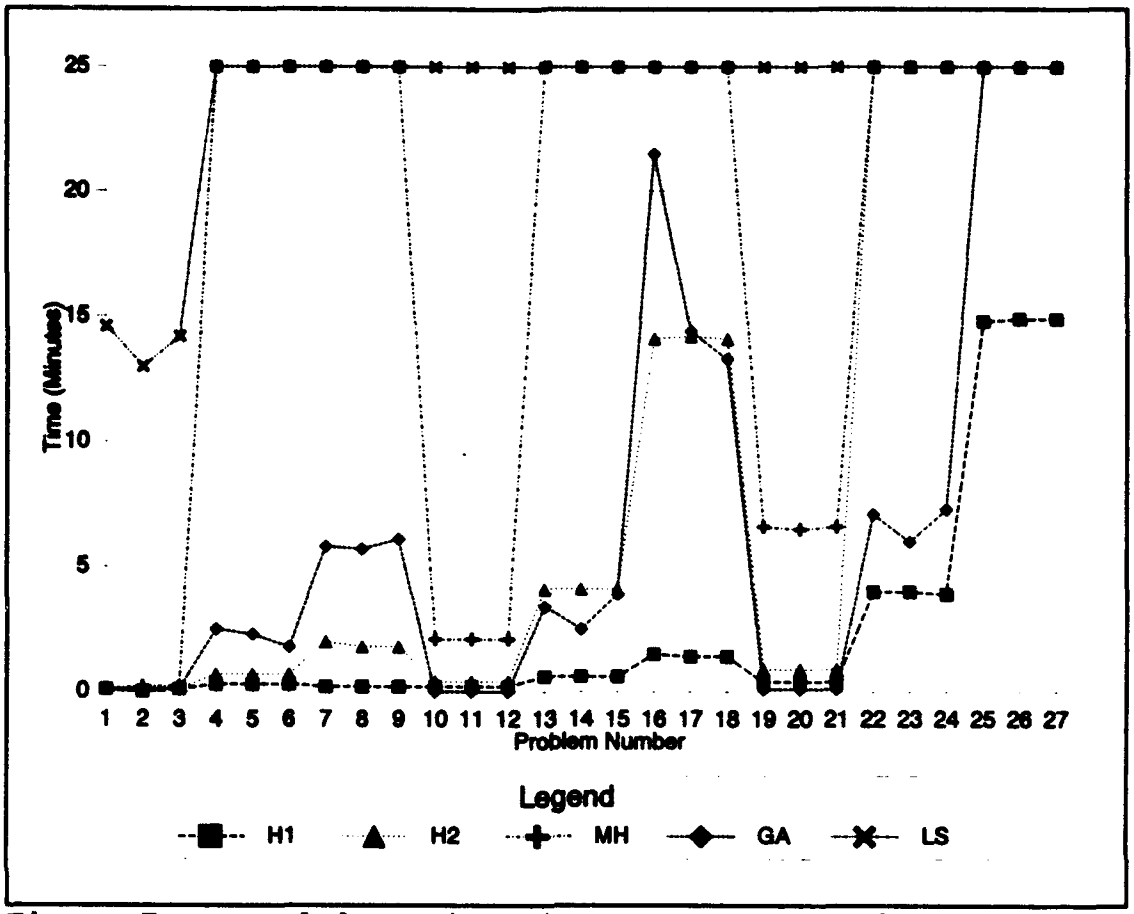

Figure 7 Expanded Running Times For Two Searcher Problems.

From Figure 7 it is transparent that only H1 and GA keep a reasonable run-time for two searchers. It is seen that H1 performs exceptionally well for its limited investment of time.

3. Three Searcher Results

Only H1 and GA exhibit reasonable run-times and quality solutions for one and two searchers and are therefore the only heuristics employed to solve three searcher problems. The 9 and 25 cell problems (problem numbers 1 to 18) are used in testing. Due to memory limitations associated with the

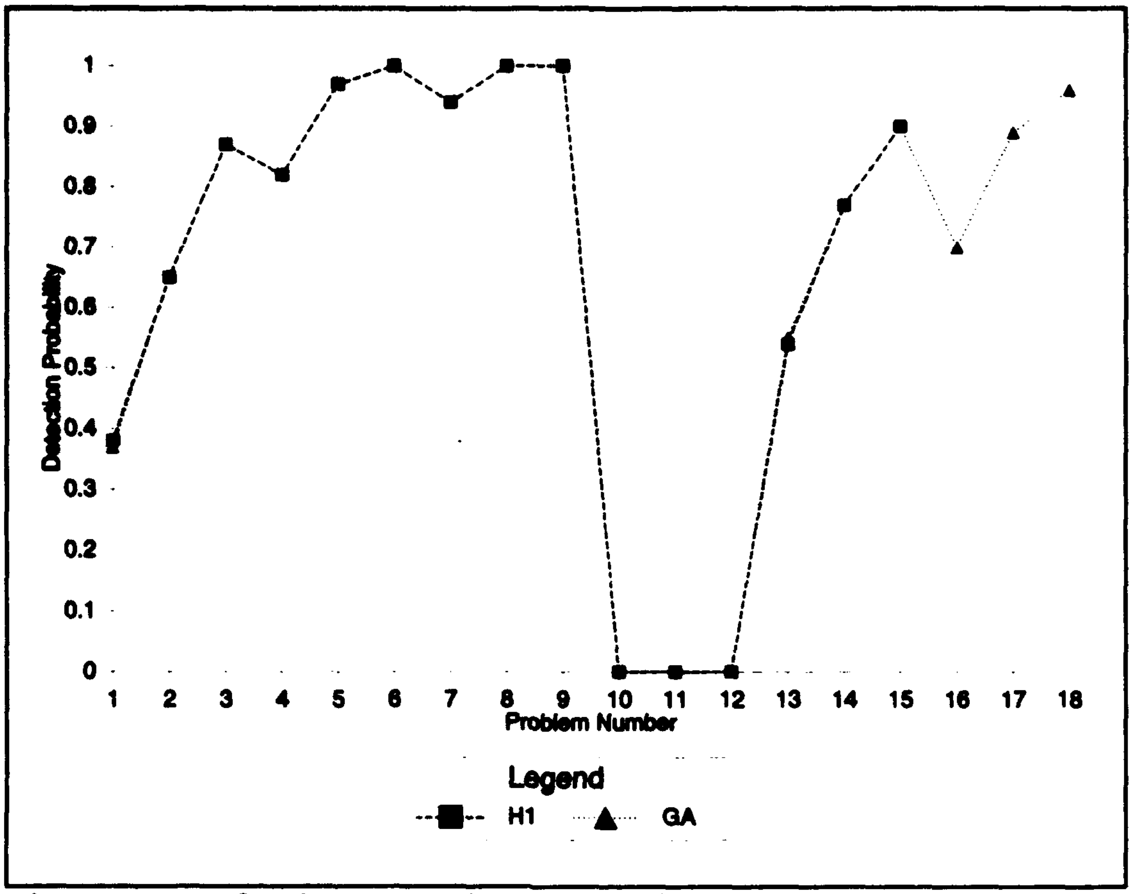

Figure 8 Solution For Three Searcher Problems.

Pascal compiler used, the 49 cell problems could not be solved without extensive reprogramming. Figure 8 presents the objective function values that GA (H1) obtains for problems 1 to 18 (1 to 15). It is clear that both heuristics obtain, for all practical purposes, the same results. Even though the optimal solution to these problems is unknown, it is reasonable to believe H1 and GA produce quality solutions due to past performance.

Figure 9 presents the run-time for both heuristics. It is clear that only GA maintains a reasonable rate of growth in its run-time with the addition of searchers.

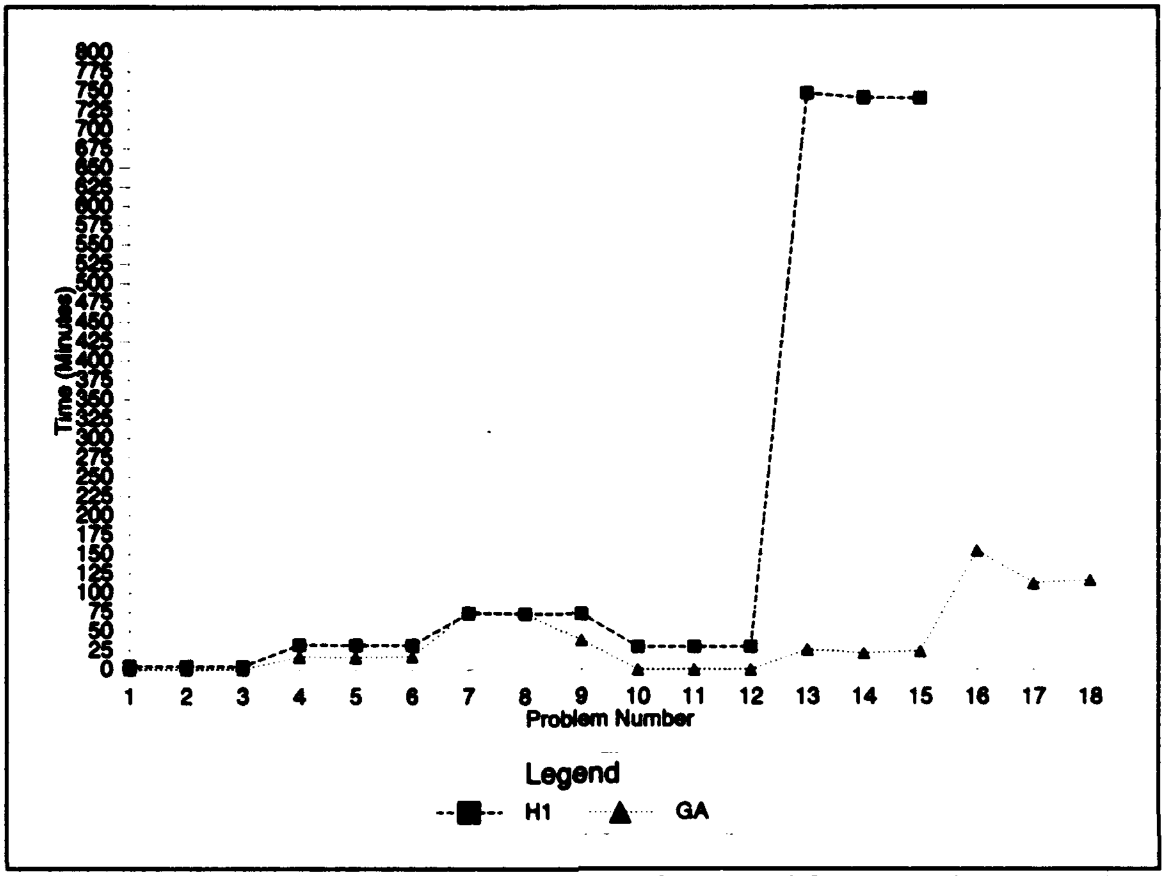

Figure 9 Run-Times For Three Searcher Problems. Observe that problems 25,26, and 27 are not solved by H1.

V. CONCLUSIONS

This thesis develops and tests effective heuristics to solve the path constrained multiple searcher problem. For more than one searcher, the time needed to guarantee an optimal solution for the problems considered is prohibitive. Heuristic H1 obtains solutions within two percent of the best known solution for each one, two, and three searcher test problems considered. For one and two searcher problems H1's solution time is less than that of other heuristics considered.

The GA heuristic performs acceptably for one and two searcher problems and highlights its ability solving three searcher problems; obtaining solutions equivalent to H1 using less than 20% of H1's run-time.

Our empirical work suggests heuristics can solve the path constrained multiple searcher problem both effectively and efficiently. Given the myriad of estimated parameters needed to model this problem, obtaining an optimal solution with respect to these estimates does not guarantee better true performance. Hence, a heuristic solution is recommended for practical applications.

A. SUGGESTIONS FOR FURTHER RESEARCH

The results of this thesis identify other related areas of research that deserve further attention. The topics follow:

- Other test scenarios that are not considered in this thesis could illuminate other traits of the problem overlooked by the proposed test cases. As an example, further research could investigate the case where the target can hide.

- The solution to the model proposed by Eagle and Yee [Ref. 2] and used in this thesis can be verified by means of simulation.

- The multiple searcher path constrained problem studied in this thesis is NP-complete, but a similar problem, solved by Eagle and Yee [Ref. 2], which assumes that the search effort is infinitely divisible, is relatively fast to solve. A study analyzing the trade-off between the effort to solve the multiple searcher problem versus Eagle's relaxed problem could identify the conditions under which each of the methods is preferred.

LIST OF REFERENCES

1. Koopman,B.O.,*A Theoretical Basis for Methods of Search and Screening*. Operations Evaluation Group Report No.56. Center For Naval Analyses, Alexandria, Virginia,1946.

2. Eagle,J.N. and Yee,J.R.,"An Optimal Branch and Bound Procedure for the Constrained Path,Moving Target Search Problem," Operations Research 38,1990.

3. Eagle,J.N. and Washburn,A.R.,"Cumulative Search-Evasion Games," *Naval Research Logistics 38 No.4*,1991.

4. Martins,G.H.,"A New Branch and Bound Procedure for Computing Optimal Search Paths," *Naval Postgraduate School Master's Thesis, 1991.*

5. Weisinger,J.R.,Trummel,K.E.,"The Complexity of the Optimal Search Path Problem," *Operations Research 34*,1986.

6. Stone,L.D.,*Theory of Optimal Search.* Academic Press,New York,1975.

7. Thomas,L.C. and Washburn,A.R.,"Dynamic Search Games," *Operations Research 39*,1991.

8. Brown,S.S.,"Optimal Search for a Moving Target in Discrete Time and Space," *Operations Research 28*,1980.

9. Washburn,A.R.,"On Search for a Moving Target," *Naval Logistics Quart. 27*, 1980.

10. Stone,L.D.,Brown,S.S.,Buemi,R.P.,and Hopkins,C.R., *"Numerical Optimization of Search for a Moving Target,"* Daniel H. Wagner,Associates Report to Office of Naval Research,1978.

11. Stewart,T.J.,"Experience with a Branch and Bound Algorithm for Constrained Searcher Motion," Plenum Press, New York,1980.

12. Stewart,T.J.,"Search for a Moving Target when Searcher motion is Restricted," *Computers & Operations Research 6*,1979.

13. Ohsumi,A.,"Optimal Search for a Markovian Target," *Naval Research Logistics 38*,1991.

14. Weisinger,J.R.,Monticino,M.G.,and Benkoski,S.J.,"A survey of the Search Theory Literature," *Naval Research Logistics 38*,1991.

15. Papadimitriou,C.H.,Steiglitz,K.,"*Combinatorial Optimization-Algorithms and Complexity.*" Prentice Hall,1982.

16. Goldberg,D.E.,"*Genetic Algorithms in Search, Optimization and Machine Learning.*", Addison-Wesley,1989.

17. Holland,J.H.,"*Adaptation in Natural and Artificial Systems.*", University of Michigan Press, 1975.

18. Koza,J.R.,"*Genetic Programming-On the Programming of Computers by Means of Natural Selection,*" M.I.T. Press,1992.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Professor Robert F. Dell, Code OR/De Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor James N. Eagle, Code OR/Er Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Alan R. Washburn, Code OR/Ws Naval Postgraduate School Monterey, California 93943-5000	1
6.	Anti-Submarine Warfare Group Attn: Mr. Dave Panetta Naval Air Warfare Development Center Warminster, Pensilvania 18974-0591	1
7.	Comando de Operações Navais Attn: Brazilian Naval Commission 4706 Wisconsin Ave., N. W. Washington, D.C. 20016	1
8.	Grupo de Desenvolvimento de Tatica Comando-em-Chefe da Esquadra 4706 Wisconsin Ave., N. W. Washington, D.C. 20016	1
9.	Brazilian Naval Commission Attn: Ilmo. Sr. Presidente 4706 Wisconsin Ave., N. W. Washington, D.C. 20016	1
10.	Centro de Analises de Sistemas Navais Attn: Brazilian Naval Commission 4706 Wisconsin Ave., N. W. Washington, D.C. 20016	1

11. LCDR Gustavo H. A. Martins 1
Attn: Brazilian Naval Commission
4706 Wisconsin Ave., N. W.
Washington, D.C. 20016

12. LCDR Almir Garnier Santos 3
Attn: Brazilian Naval Commission
4706 Wisconsin Ave., N. W.
Washington, D.C. 20016